

## DESENVOLVIMENTO DE UM SOFTWARE PARA AUTOMAÇÃO DE PROCESSOS UTILIZANDO LÓGICA FUZZY

BRUNO SIELLY J. COSTA\*, CLAUBER BEZERRA†, LUIZ AFFONSO GUEDES‡

\**Instituto Federal de Educação Ciência e Tecnologia do Rio Grande do Norte  
Campus Natal - Zona Norte  
Natal, RN, Brasil*

†*Instituto Federal de Educação Ciência e Tecnologia do Rio Grande do Norte  
Campus Santa Cruz  
Santa Cruz, RN, Brasil*

‡*Universidade Federal do Rio Grande do Norte  
Departamento de Engenharia de Computação e Automação  
Natal, RN, Brasil*

Emails: [bruno.costa@ifrn.edu.br](mailto:bruno.costa@ifrn.edu.br), [clauber.bezerra@ifrn.edu.br](mailto:clauber.bezerra@ifrn.edu.br), [affonso@dca.ufrn.br](mailto:affonso@dca.ufrn.br)

**Abstract**— This paper presents the design and implementation of a software that allows modeling, building and editing fuzzy inference systems. The software was developed using Java language and it provides a simple, functional and very intuitive user interface and also has a freeware license. The developed system allows use the inference rules created in industrial process control, supporting communication through OPC protocol. The main features of developed system are flexibility, robustness, reliability and ease of expansion.

**Keywords**— Fuzzy logic, Software Engineering, Process Automation.

**Resumo**— Este artigo apresenta o projeto e implementação de um software que permite a modelagem, construção e edição de sistemas de inferência fuzzy. O software, que foi desenvolvido utilizando a linguagem Java, apresenta uma interface simples, funcional e bastante intuitiva, além de possuir uma licença freeware. O ambiente desenvolvido permite ainda a aplicação das regras de inferência criadas no controle de processos industriais, utilizando o padrão OPC para comunicação com os equipamentos de campo. As principais características do sistema desenvolvido são flexibilidade, robustez, confiabilidade e facilidade de expansão.

**Palavras-chave**— Lógica fuzzy, Engenharia de software, Automação de processos.

### 1 Introdução

Em muitas aplicações envolvendo a automação de processos industriais, existe a necessidade de classificar o estado atual de uma variável para que esta aplicação possa tomar decisões a respeito dos próximos passos a serem executados por ela.

A lógica fuzzy, proposta em 1965 por Lofti Zadeh (Zadeh, 1965), permite o tratamento de informações vagas, imprecisas e ambíguas. Ela é uma ferramenta que vem se mostrando adequada a solução de problemas na área de automação de processos, principalmente que envolvam a tomada de decisões, pois proporciona uma análise de dados semelhante ao raciocínio humano.

Além disso, a utilização da lógica fuzzy no controle de processos é de fácil implementação, não necessitando de modelos matemáticos complexos. A lógica fuzzy utiliza apenas um conjunto simples de regras, que são baseadas na experiência dos operadores ou especialistas em determinado processo industrial.

A utilização da lógica fuzzy no controle de processos foi proposta pela primeira vez por Ebrahim Mamdani no controle de uma máquina a vapor (Mamdani, 1974). Mamdani já havia tentado controlar esse processo com outros tipos de controladores, porém todos sem sucesso.

Anos depois, estudos mostraram que um controlador fuzzy era mais eficiente do que um controlador convencional no controle dos processos de aceleração, frenagem e parada dos trens no metrô de Sendai no Japão (Yasunobu et al., 1983). O controlador fuzzy proporcionava aceleração e frenagem mais suaves, paradas mais precisas nas plataformas além de reduzir o consumo de energia.

Além dessas, várias outras aplicações envolvendo lógica fuzzy surgiram nos últimos anos, como controle de eletrodomésticos, aeronaves não tripuladas, elevadores, dentre outras (Pinto, 2010). Neste aspecto, existe uma grande demanda por ferramentas de software que possibilitem aos usuários a aplicação dos conceitos inerentes à lógica fuzzy em processos de automação industrial. Na literatura podemos encontrar alguns toolboxes implementados para trabalhar com lógica fuzzy.

Hall e Hathaway (Hall and Hathaway, 1996) descrevem um toolbox de lógica fuzzy integrado a plataforma *MathWorks Matlab*, com uma interface gráfica amigável para construção e edição de sistemas de inferência fuzzy. Chuan (Chuan et al., 2004) apresenta um toolbox de lógica fuzzy baseado na linguagem do *Scilab* e o mostra como uma alternativa gratuita ao software da *MathWorks*.

A proposta deste trabalho é apresentar um

completo e poderoso software para manipulação de regras de inferência da lógica fuzzy. Este software, chamado JFuzZ, pode ser utilizado em diversas aplicações que envolvam a automação de processos industriais, pois ele possui uma estrutura que permite a comunicação em tempo real com uma grande variedade de equipamentos de campo, possibilitando assim a aplicação das regras criadas em diversos tipos de processos.

## 2 Lógica Fuzzy

A lógica clássica classifica um dado como pertencente ou não a um determinado conjunto, ou seja, a classificação é feita com base em apenas dois possíveis valores: verdadeiro ou falso. Dessa forma, uma variável de processo possui apenas dois estados pré-definidos e complementares, por exemplo, o nível de um tanque pode estar alto ou baixo, a temperatura de uma caldeira pode estar quente ou fria, uma válvula pode estar aberta ou fechada, etc.

Porém, muitas vezes esses dois estados não são suficientemente representativos para determinada aplicação. Por exemplo, imagine que em um sistema de controle de nível definimos que o nível de um tanque está alto quando ele for igual ou superior a 5m e baixo caso contrário. Isso nos levaria a concluir que 4,99m é um nível baixo e 5m é um nível alto, onde na verdade esses dois valores são muito próximos e deveriam ser tratados pelo controlador de forma semelhante.

Além disso, podemos também concluir, de acordo com esses critérios, que um nível de 0m e outro de 4,99m serão tratados da mesma forma pelo controlador pois os dois são considerados baixos, porém isso não deveria ocorrer pois os valores representam situações bem diferentes do processo.

A teoria dos conjuntos fuzzy, desenvolvida por Lofti Zadeh (Zadeh, 1965), permite tratar a incerteza existente em um conjunto de dados. Dessa forma, ela se torna uma ferramenta adequada ao tratamento de dados que possuam uma certa imprecisão, se aproximando mais do raciocínio humano, possibilitando assim automatizar diversas tarefas de automação industrial.

Esses conjuntos serviram de base para o desenvolvimento da lógica fuzzy, que permite classificar uma informação como sendo parcialmente pertencente a um conjunto. De acordo com a teoria dos conjuntos fuzzy, uma informação pode pertencer apenas parcialmente a um conjunto, além de que ela pode pertencer a mais de um conjunto simultaneamente. A classificação é feita de acordo com uma ponderação entre os conjuntos a que ela pertence.

A pertinência indica o quanto um elemento pertence a um determinado conjunto. Para medir esse grau são utilizadas as funções de pertinência, como as mostradas na figura 1. Nessa figura temos

vários conjuntos fuzzy utilizados para classificar o nível de um tanque. Observe que o nível de 4,7m pertence tanto ao conjunto baixo, como ao conjunto normal, onde em cada conjunto esse nível possui um grau de pertinência.

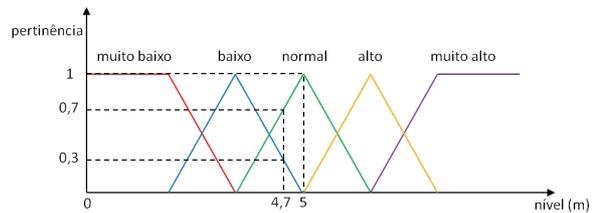


Figura 1: Conjuntos fuzzy.

### 2.1 Sistema de inferência fuzzy

Os sistemas de inferência fuzzy são ferramentas computacionais que utilizam uma série de conceitos da lógica fuzzy para tomar decisões automaticamente sem intervenções humanas. A estrutura básica de um sistema de inferência é mostrada na figura 2.

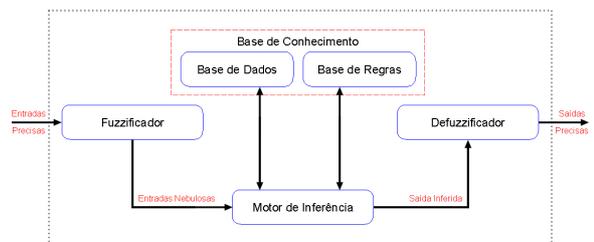


Figura 2: Sistema de inferência fuzzy.

Essa estrutura é formada por quatro elementos que interagem entre si para a partir de um conjunto de valores de entrada, gerar um conjunto de valores de saída, que quantificam a decisão tomada pelo sistema. São eles:

- **Fuzzificador:** esse elemento converte um conjunto de valores de entrada em valores fuzzy. Para isso, ele utiliza funções de pertinência pré-definidas, verificando o grau de pertinência para cada valor de entrada. Essas funções podem ser de vários tipos como triangulares, trapezoidais e gaussianas.
- **Base de conhecimento:** é composta por uma base de dados e uma base de regras de inferência. A base de regras armazena um conjunto de regras do tipo Se-Então, formadas por um antecedente (Se) e um conseqüente (Então). Essas regras são criadas com base nos valores da base de dados e são utilizadas para produzir a saída do sistema.
- **Motor de inferência:** utiliza a base de conhecimento para procurar respostas as entradas do sistema. Ele realiza comparações dos

valores fuzzy de entrada com os antecedentes das regras e combina os seus consequentes para gerar as saídas fuzzy do sistema.

- **Defuzzificador:** realiza o processo contrário da fuzzyficação. Ele transforma os valores fuzzy inferidos em valores de saída. Para isso podem ser utilizados os métodos do centróide, do bissetor, média dos máximos, primeiro máximo e último máximo.

O software proposto neste trabalho permite a criação de sistemas de inferência baseados nos modelos propostos por Mamdani (Mamdani and Assilian, 1975) e Takagi-Sugeno (Takagi and Sugeno, 1985).

### 3 OPC

O padrão OPC (*OLE for Process Control*) é um protocolo de comunicação baseado nas tecnologias OLE/DCOM da Microsoft (OPC, 2003). A ideia desse protocolo é padronizar o formato com que os dados de equipamentos de campo são disponibilizados para os softwares de automação industrial. Com isso, um software pode enviar ou receber dados para um equipamento independente do seu fabricante.

A arquitetura de funcionamento de uma aplicação baseada no padrão OPC possui um software chamado de servidor OPC. Esse servidor geralmente é disponibilizado pelo fabricante do equipamento e se comunica com o PLC (*Programmable Logic Controller*) através do protocolo proprietário do fabricante. O servidor disponibiliza os dados do PLC para as aplicações cliente utilizando o padrão OPC de forma que um programa cliente pode acessar dados de diferentes fabricantes utilizando uma interface de comunicação única.

A figura 3 ilustra essa ideia, onde um conjunto de aplicações utilizam interfaces OPC para acessar dados de diferentes servidores OPC através de uma rede de computadores.

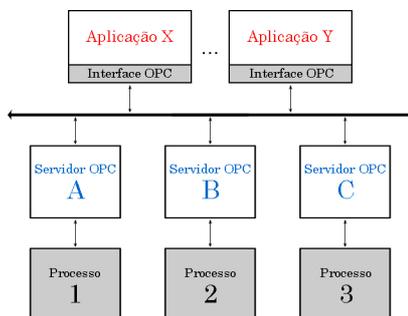


Figura 3: Arquitetura OPC-DA.

## 4 Arquitetura do sistema

O JFuzZ foi desenvolvido utilizando a linguagem Java e possui uma série de ferramentas que per-

mitem a criação e edição de sistemas de inferência fuzzy. Uma vez criados e modelados, esses sistemas podem ser utilizados no controle de processos industriais em geral, utilizando o protocolo de comunicação OPC-DA.

O sistema foi desenvolvido com base numa arquitetura modular, onde cada módulo é responsável por um conjunto de funcionalidades bem definidas do sistema. Esses módulos interagem entre si, de forma que cada módulo disponibiliza as suas funcionalidades a outros módulos. A utilização dessa arquitetura facilita o desenvolvimento e a manutenção do software. A figura 4 ilustra os cinco módulos do JFuzZ, como também a interação entre eles. Cada um desses módulos será descrito a seguir.

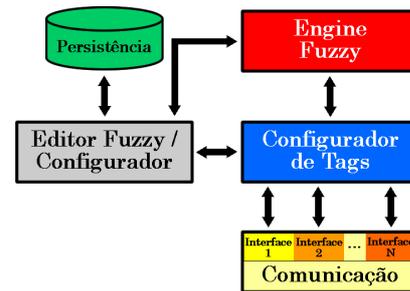


Figura 4: Módulos do JFuzZ.

### 4.1 Módulo de edição fuzzy / configurador

Esse módulo configura uma série de informações necessárias ao sistema de inferência, tais como o tipo e o nome das variáveis, definição de quais variáveis de inferência são de entrada e quais são de saída, funções de pertinência, ranges, métodos de inferência, etc. Após realizadas essas configurações, esse módulo disponibiliza essas informações ao módulo de *Engine Fuzzy*.

### 4.2 Módulo de Engine Fuzzy

Esse é o módulo responsável por realizar o processo de inferência propriamente dito. Ele coleta as informações do sistema de inferência, disponibilizadas pelo módulo de configuração, e através dos parâmetros coletados, realiza as operações matemáticas sobre os valores de entrada, gerando um conjunto de valores de saída. O *Engine Fuzzy* implementa os principais métodos encontrados na literatura para sistemas de inferência fuzzy, os quais são mostrados na figura 5.

### 4.3 Módulo de Comunicação

Esse módulo é responsável pela troca de informações entre o sistema de inferência e o processo. Ele é composto por uma lista de interfaces de comunicação industrial, sendo cada uma delas capaz de

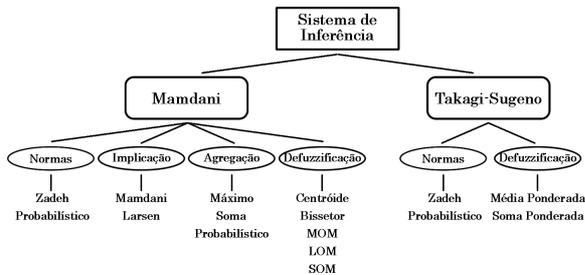


Figura 5: Métodos implementados no sistema de inferência.

comunicar-se com um protocolo de comunicação diferente.

Esse módulo é projetado de forma que o sistema de inferência não precisa preocupar-se com os detalhes de comunicação com o processo. Os dados do processo são disponibilizados ao sistema de inferência de forma transparente, ou seja, independente da interface de comunicação utilizada com o processo. Assim, esse módulo entrega os dados num formato genérico para que eles sejam utilizados pelo sistema de inferência.

Uma das interfaces desenvolvidas no sistema é a OPC, pois ela permite a comunicação com uma grande quantidade de equipamentos de campo que utilizam esse protocolo, já que o mesmo é bastante adotado atualmente na indústria.

#### 4.4 Módulo de Configuração de Tags

O módulo configurador de tags converte as variáveis disponibilizadas pelo módulo de comunicação em várias de inferência. Esse módulo permite associar uma variável de inferência a uma expressão aritmética que combine diferentes variáveis de processo. Essa expressão é criada utilizando uma série de operações aritméticas disponibilizadas por esse módulo.

Para exemplificar a utilidade desse módulo, imagine que seja necessário utilizar uma variável de erro no sistema de inferência. Geralmente essa variável não está explicitamente disponível no processo, mas pode ser calculada através da diferença entre o *setpoint* e o valor lido de uma das variáveis do processo.

#### 4.5 Módulo de persistência

Esse módulo é responsável pelo armazenamento e recuperação de todas as informações do sistema de inferência fuzzy. Durante o projeto do JFuzZ optou-se pela utilização de arquivos de texto para realizar esse armazenamento.

Todos os parâmetros de configuração dos sistemas de inferência criados podem ser gravados em um arquivo estrutural, de extensão “.FUZ”. Os dados armazenados nos arquivos podem ser divididos em quatro grupos distintos, conforme mostra a figura 6. São eles:

- **Dados Básicos:** são os dados referentes as configurações básicas do sistema de inferência, tais como tipos de normas, métodos de implicação, agregação e defuzzificação.
- **Entrada:** armazena informações relacionadas as variáveis de entrada do sistema, tais como nomes, ranges e informações de servidores e variáveis do processo. Também armazena informações de todas as funções de pertinência associadas a cada variável de entrada.
- **Saída:** são os dados a respeito das variáveis de saída. Guarda informações tais como nomes, ranges e informações de servidores e variáveis do processo. Além disso, este bloco armazena as informações de todas as funções de pertinência associadas a cada variável de saída.
- **Regras:** dados a respeito das regras de inferência definidas no sistema fuzzy.

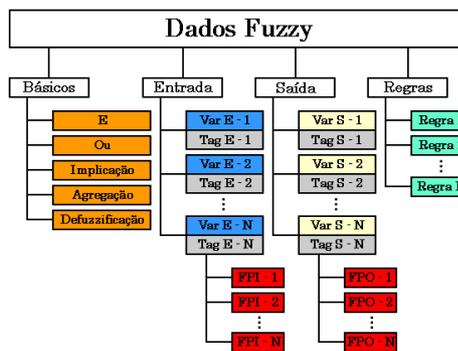


Figura 6: Estrutura de dados do JFuzZ

## 5 Funcionalidades do Sistema

O sistema JFuzZ foi desenvolvido utilizando sete ambientes gráficos, que disponibilizam as funcionalidades do sistema aos usuários. Cada uma desses ambientes e suas funcionalidades serão descritas nas próximas subseções.

### 5.1 Editor básico de sistemas de inferência fuzzy

O editor de sistemas de inferência permite configurar o número de variáveis de entrada e saída, os nomes das variáveis, os links de variáveis do sistema com variáveis de processo dentre outras configurações básicas do sistema de inferência. Esse ambiente é mostrado na figura 7, onde os campos destacados são:

1. **Barra de títulos:** exibe o tipo do sistema de inferência e o nome do arquivo aberto atualmente.

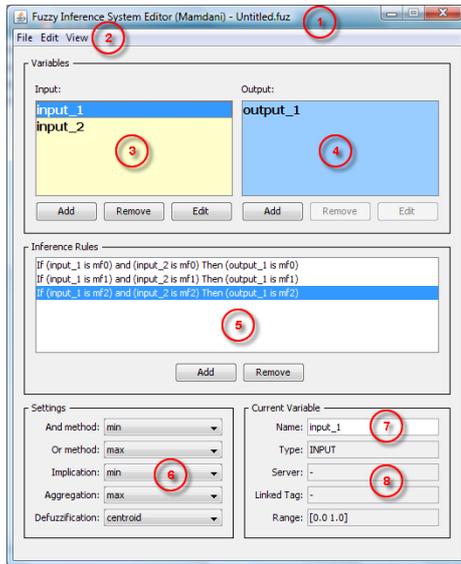


Figura 7: Ambiente de edição de sistemas de inferência.

2. **Barra de menus:** permite abrir, salvar e editar os sistemas de inferência, bem como acessar os outros ambientes gráficos.
3. **Variáveis de entrada:** exibe as variáveis de entrada do sistema, permitindo a adição, remoção e edição de variáveis.
4. **Variáveis de saída:** exibe as variáveis de saída do sistema, permitindo a adição, remoção e edição de variáveis.
5. **Regras de inferência:** exibe as regras de inferência do tipo Se-Então do sistema, permitindo a adição e remoção de regras.
6. **Configurações do sistema:** seleção de métodos e ajustes das funções de inferência.
7. **Nome da variável:** mostra e permite modificar o nome de uma variável de entrada ou saída do sistema.
8. **Campos informativos:** exibem outras informações sobre a variável, tais como o seu tipo, a *tag* à qual ela está ligada, seus limites de operação, etc.

### 5.2 Editor de funções de pertinência

Permite a visualização e edição de todas as funções de pertinência associadas a todas as variáveis de entrada e saída do sistema. Permite também modelar as funções de acordo com os tipos disponíveis e com os parâmetros de configuração e de range.

A figura 8 ilustra esse ambiente, com os seguintes campos destacados:

1. **Barra de menus:** permite adicionar, remover e salvar funções de pertinência do sistema de inferência.

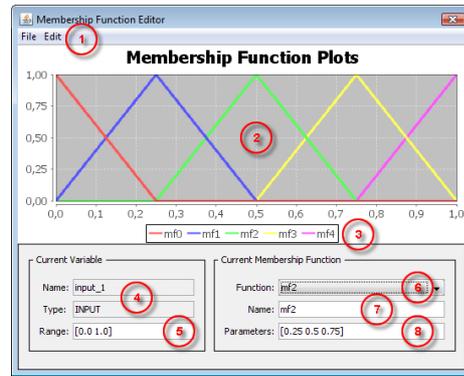


Figura 8: Ambiente de edição de funções de pertinência.

2. **Gráfico principal:** este gráfico exibe todas as funções de pertinência da variável selecionada.
3. **Legenda:** relaciona as funções mostradas no gráfico aos seus respectivos nomes.
4. **Nome e tipo da variável:** exibe o nome e tipo da variável selecionada.
5. **Range:** define os limites mínimo e máximo de valores que a variável poderá atingir.
6. **Função atual:** seleciona uma das funções de pertinência existentes.
7. **Nome da função:** este campo de texto é usado para mostrar e modificar o nome da função de pertinência selecionada.
8. **Parâmetros:** configuração dos parâmetros de modelagem do comportamento da função. O número de parâmetros depende do tipo da função de pertinência.

### 5.3 Editor de regras de inferência

Baseado nas descrições das variáveis de entrada e saída definidas no editor básico, esse ambiente permite criar regras de inferência de forma simples, bastando apenas selecionar e associar cada variável a um valor fuzzy. Essa associação é opcional, quando nenhum valor é selecionado, a variável em questão é ignorada para aquela regra na etapa de inferência. A figura 9 ilustra esse ambiente, destacando os seguintes campos:

1. **Variáveis de entrada:** campos com os nomes de todas as variáveis de entrada do sistema para a criação da regra de inferência.
2. **Variáveis de saída:** campos com os nomes de todas as variáveis de saída do sistema para a criação da regra de inferência.
3. **Valores fuzzy:** campo de seleção com todos os valores fuzzy disponíveis para cada variável.

4. **Conector lógico:** conector de ligação das associações das variáveis de entrada com os seus respectivos valores fuzzy.

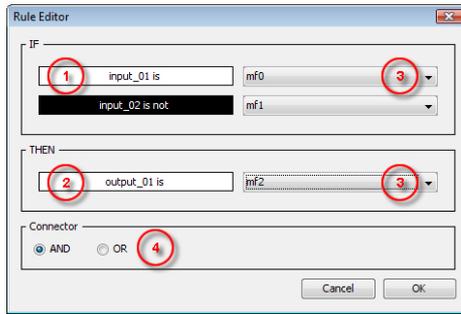


Figura 9: Ambiente de edição de regras de inferência.

#### 5.4 Visualizador de regras

Neste ambiente são exibidos os antecedentes e consequentes de cada regra. Para cada uma delas, os valores fuzzy selecionados para as variáveis de entrada e saída são ilustrados na forma de gráficos, de acordo com as funções de pertinência associadas. Além disso, o ambiente exibe a forma que representa a operação de agregação, calculando e identificando os valores para as variáveis de saída, de acordo com as entradas do sistema e dos parâmetros gerais de configuração. A figura 10 ilustra esse ambiente, onde os campos em destaque são:

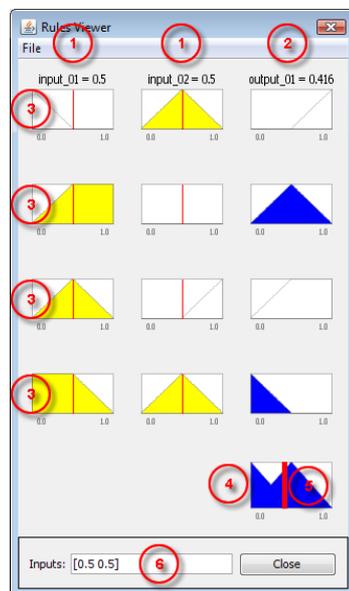


Figura 10: Ambiente de visualização de regras de inferência.

1. **Antecedentes:** cada coluna gráfica amarela mostra como a variável de entrada é usada em cada regra. Os valores para cada variável de entrada são mostrados no topo da coluna.

2. **Consequentes:** cada coluna gráfica azul mostra como a variável de saída é usada em cada regra. Os valores para cada variável de saída são mostrados no topo da coluna.

3. **Regras de inferência:** cada linha de gráficos representa uma regra de inferência.

4. **Forma de saída agregada:** o último gráfico da tela mostra como a saída de cada regra é combinada para formar a saída agregada.

5. **Defuzzificação:** a linha indica o valor de saída do defuzzificador para a forma de saída gerada.

6. **Entradas:** caixa de texto para inserção de valores de entradas para o sistema.

#### 5.5 Módulo de controle

O módulo de controle é o ambiente de execução dos sistemas de inferência fuzzy aplicados a um determinado processo industrial. Após criado no editor básico de sistemas de inferência, tal sistema fuzzy deve ser carregado no módulo de controle para início da execução.

Nesse ambiente, o operador pode monitorar o processo de controle através de informações numéricas e gráficos de tendências em tempo real. Além disso, o operador pode alterar os valores de referência do sistema a qualquer momento, sendo o controlador capaz de ajustar-se para seguir a nova referência. A figura 11 ilustra esse ambiente, onde os campos destacados são:

1. **Configuração do sistema de inferência atual:** botão de acesso ao editor básico de sistemas de inferência. A partir dele, o editor carregará o sistema de inferência atualmente utilizado no módulo de controle. Se nenhum sistema de inferência fuzzy encontra-se carregado, um novo será criado.

2. **Carregamento de sistema de inferência existente:** botão de acesso ao editor básico de sistemas de inferência. A partir dele, o módulo de controle carregará um sistema de inferência previamente definido, a partir de um arquivo “.FUZ”.

3. **Informações básicas:** informações sobre o sistema de inferência carregado.

4. **Setpoint:** campo de definição da referência do sistema de controle.

5. **Dados de variáveis:** valores de tempo-real das variáveis do sistema.

6. **Período de atualização:** intervalo de atualização do sistema de controle, dado em milissegundos. Pode ser limitado à taxa de atualização do meio de comunicação.

7. **Execução da tarefa:** inicia e finaliza a execução da tarefa de controle, de acordo com o período de atualização definido.
8. **Variáveis ativas:** lista de variáveis ativas, e passíveis de monitoramento.
9. **Geração de gráficos:** a partir de seleção realizada na lista das variáveis, gera um gráfico de tendência com os valores das variáveis, atualizado a cada ciclo de controle. Vários gráficos podem ser gerados, com diferentes combinações de variáveis.

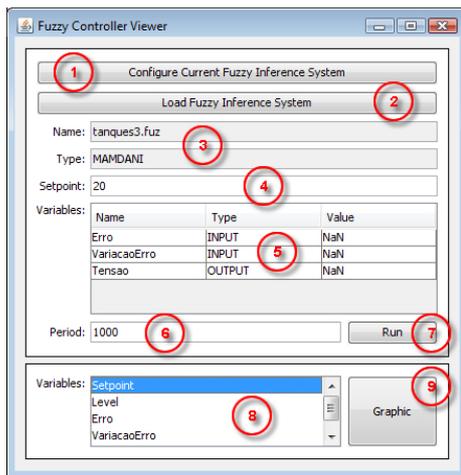


Figura 11: Ambiente de visualização do controlador fuzzy.

### 5.6 Registrador de Variáveis

Este ambiente, que é mostrado na figura 12, permite ao usuário registrar uma variável de inferência no sistema. Ele possibilita que seja feita uma busca por *tags* em algum servidor OPC conectado na rede. Essa busca é facilitada com o auxílio de uma máscara de busca, que permite filtrar os resultados da busca. Além disso, nesse ambiente é disponibilizado uma calculadora de *tags*, que permite associar a variável de inferência a uma expressão aritmética envolvendo uma ou mais *tags* do processo.

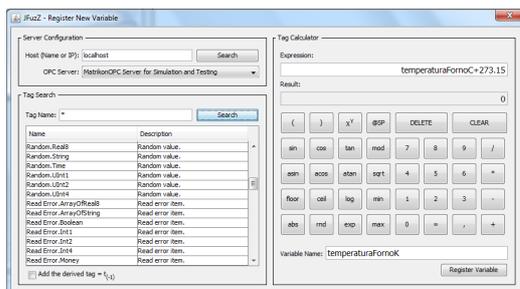


Figura 12: Ambiente para edição de *tags* calculadas.

## 6 Validação

Para validar o JFuzZ, aplicamos o mesmo no controle de um processo real. O processo escolhido foi o controle de nível de um líquido em um tanque. Para isso, utilizamos uma planta experimental composta por dois tanques acoplados da Quanser (Quanser, 2004).

Essa planta consiste em dois tanques acoplados verticalmente com um reservatório de água abaixo deles. Uma bomba envia água do reservatório para o tanque superior, que por sua vez possui um orifício na sua base que permite o fluxo da água para o tanque inferior. Além disso, o tanque inferior também possui um orifício que permite a saída da água desse tanque de volta para o reservatório.

Neste experimento utilizaremos o JFuzZ para controlar o nível do tanque superior da planta, onde esse nível pode variar de 0cm a 30cm. A figura 13 mostra a arquitetura do sistema de controle implementado. O JFuzZ se comunica com um servidor OPC, que se comunica com um CLP ligado a um sensor que indica o nível do tanque.

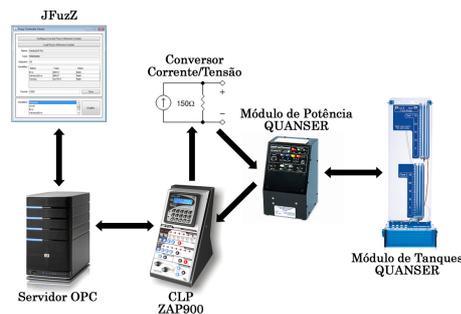


Figura 13: Arquitetura do controlador implementado.

Para controlar esse processo utilizamos a estratégia de controle fuzzy multiestágio descrita por Costa et al. (2012). Nessa estratégia o tanque é dividido em três partes: superior, central e inferior, onde cada uma dessas partes possui um sistema de controle fuzzy diferente.

Utilizando as funcionalidades do JFuzZ criamos um sistema de inferência baseado no modelo de Mamdani para controlar o processo. Nesse sistema as variáveis de entrada são o erro e a variação do erro do nível do tanque. Já a variável de saída é o sinal de controle, que é a tensão aplicada a bomba. Para cada uma dessas variáveis criamos um conjunto de valores fuzzy e em seguida três tabelas de associação entre os valores de entrada (Costa et al., 2012).

Criado o conjunto de regras, iniciamos um experimento com o *setpoint* em 15cm e após a estabilização do nível do tanque alteramos o *setpoint* para 20cm. Por fim, alteramos mais uma vez o *setpoint* para 10cm. Com isso, verificamos que para todos os *setpoints* utilizados o controlador se

comportou de maneira satisfatória. A figura 14 mostra os resultados obtidos utilizando o JFuzZ no controle do processo descrito.

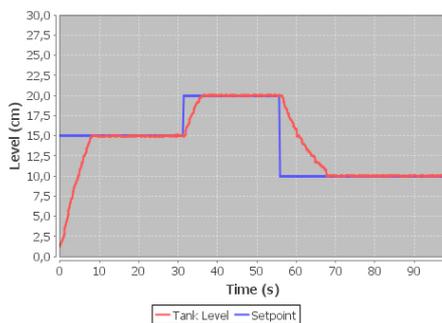


Figura 14: Controle do nível do tanque com setpoint variável.

## 7 Conclusão

Neste artigo apresentamos um software para criação e edição de sistemas de inferência fuzzy. Esse software permite criar sistemas de inferência baseados nos modelos de Mamdani e Takagi-Sugeno. O software disponibiliza um conjunto de ambientes gráficos bastante intuitivos que facilitam a criação desses sistemas de inferência.

Além disso, o software é gratuito e desenvolvido utilizando a linguagem Java, o que o torna independente de plataforma. O software completo está disponível no endereço <http://www.dca.ufrn.br/bruno/jfuzz.rar>.

Esse software permite a utilização de uma série de interfaces de comunicação, o que permite a aplicação do sistema de inferência a um processo industrial. No desenvolvimento deste artigo, utilizamos uma interface de comunicação OPC, pois se trata de um padrão de comunicação amplamente utilizado nos dias de hoje. Porém, devido à estrutura modular do sistema, outras interfaces de comunicação podem ser facilmente agregadas no futuro.

## Agradecimentos

Agradecemos ao Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte, campus Natal Zona Norte e campus Santa Cruz, e ao Departamento de Engenharia de Computação e Automação da Universidade Federal do Rio Grande do Norte por disponibilizar a estrutura necessária ao desenvolvimento desta pesquisa.

## Referências

Chuan, F., Zengqi, S. and Ling, S. (2004). Design and implementation of scilab fuzzy logic toolbox, *IEEE International Symposium on Computer Aided Control Systems Design*.

Costa, B. S. J., Bezerra, C. G. and de Oliveira, L. A. H. G. (2012). A multistage fuzzy controller: Toolbox for industrial applications, *IEEE International Conference on Industrial Technology*.

Hall, L. and Hathaway, R. (1996). Fuzzy logic toolbox - software review, *IEEE Transactions on Fuzzy Systems* 4(1).

Mamdani, E. H. (1974). Application of fuzzy algorithms for simple dynamic plant, *Proceedings of Institute of Electrical Engineering* 121: 1585–1588.

Mamdani, E. H. and Assilian, S. (1975). An experiment in linguistic synthesis with a fuzzy logic controller., *International Journal of Man-Machine Studies* 7(1): 1–13.

OPC (2003). Opc data access custom interface specification 3.0, *Technical report*, Opc Foundation.

Pinto, R. L. (2010). *Aplicação de um sistema especialista fuzzy para redução de manobras de dispositivos shunts chaveados automaticamente por um compensador estático*, Dissertação de mestrado, COPPE, Universidade Federal do Rio de Janeiro.

Quanser (2004). Coupled tanks user manual, *Technical report*, Quanser.

Takagi, T. and Sugeno, M. (1985). Fuzzy identification of system and its applications to modeling and control, *IEEE Transactions on Systems, Man and Cybernetics* pp. 116–132.

Yasunobu, S., Miyamoto, S. and Ihara, H. (1983). Fuzzy control for automatic train operation system, *Proceedings of 4th IFAC/IFIP/IFORS - International Congress on Control in Transportation Systems* pp. 33–39.

Zadeh, L. (1965). Fuzzy sets, *Information Control* 8: 338–353.